The DOM Scripting Toolkit: jQuery

Remy Sharp
Left Logic

Why JS Libraries?

- DOM scripting made easy

Why JS Libraries?

- DOM scripting made easy
- Cross browser work done for you

Why JS Libraries?

- DOM scripting made easy
- Cross browser work done for you
- Puts some fun back in to coding

Why jQuery?

- Lean API, makes your code smaller

Why jQuery?

- Lean API, makes your code smaller
- Small (15k gzip'd), encapsulated, friendly library - plays well!

Why jQuery?

- Lean API, makes your code smaller
- Small (15k gzip'd), encapsulated, friendly library - plays well!
- Plugin API is *really* simple

Why jQuery?

- Lean API, makes your code smaller
- Small (15k gzip'd), encapsulated, friendly library - plays well!
- Plugin API is *really* simple
- Large, active community

Why jQuery?

- Lean API, makes your code smaller
- Small (15k gzip'd), encapsulated, friendly library - plays well!
- Plugin API is *really* simple
- Large, active community
- Big boys use it too: Google, BBC, Digg, Wordpress, Amazon, IBM.

What's in jQuery?

- Selectors & Chaining
- DOM manipulation
- Events
- Ajax
- Simple effects

Selectors

```
$('#emails a.subject');
```

Selectors

- “Find something, do something with it”

Selectors

- “Find something, do something with it”
- The dollar function

Selectors

- "Find something, do something with it"
- The dollar function
- CSS 1-3 selectors at the core of jQuery

Selectors

- "Find something, do something with it"
- The dollar function
- CSS 1-3 selectors at the core of jQuery
- Custom selectors

CSS Selectors

```
$('div')
```

CSS Selectors

```
$('div')

$('div.foo')
```

CSS Selectors

```
$('div')

$('div.foo')

$('a[type="application/pdf"]')
```

CSS Selectors

```
$('div')

$('div.foo')

$('a[type="application/pdf"]')

$('tr td:first-child')
```

Custom Selectors

```
$('div:visible')
```

Custom Selectors

```
$('div:visible')

$(':animated')
```

Custom Selectors

```
$('div:visible')

$(':animated')

$(':input')
```

Custom Selectors

```
$('div:visible')

$(':animated')

$(':input')

$('tr:odd')
```

Selector Performance

```
$('#email') // same as getElementById
```

Selector Performance

```
$('#email') // same as getElementById

$('.email') // slower on a big DOM
```

Selector Performance

```
$('#email') // same as getElementById
$('.email') // slower on a big DOM
// using context
$('#emails .subject')
$('.subject', this)
```

Selector Performance

```
$('#email') // same as getElementById
$('.email') // slower on a big DOM
// using context
$('#emails .subject')
$('.subject', this)

// Caching
var subjects = $('#emails .subject');
```

Chaining

```
$('#emails .subjects')
    .click()
    .addClass('read');
```

Chaining

- jQuery returns itself *

* except when requesting string values, such as .css() or .val()

Chaining

- jQuery returns itself *
- We can use the selector once, and keep manipulating

* except when requesting string values, such as .css() or .val()

Chaining

- jQuery returns itself *
- We can use the selector once, and keep manipulating
- Can reduce size of our code

* except when requesting string values, such as .css() or .val()

Chaining in Action

```
var image = new Image();

$(image)
  .load(function () {
    // show new image
  })
  .error(function () {
    // handle error
  })
  .attr('src', '/path/to/large-image.jpg');
```

More Chaining

```
// simple tabs

$('a.tab').click(function () {
  $(tabs)
    .hide()
    .filter(this.hash)
      .show();
});
```

Collections

```
$('#emails .subjects').each(fn)
```

Collections

- .each(fn)
  Iterates through a collection applying the method

Collections

- .each(fn)
  Iterates through a collection applying the method
- .map(fn)
  Iterates through collection, returning array from fn

Working the DOM

```
$(this).addClass('read').next().show();
```

DOM Walking

- Navigation: children, parent, parents, siblings, next, prev

DOM Walking

- Navigation: children, parent, parents, siblings, next, prev
- Filters: filter, find, not, eq

DOM Walking

- Navigation: children, parent, parents, siblings, next, prev
- Filters: filter, find, not, eq
- Collections: add, end

DOM Walking

- Navigation: children, parent, parents, siblings, next, prev
- Filters: filter, find, not, eq
- Collections: add, end
- Tests: is

DOM Walking

- Navigation: children, parent, parents, siblings, next, prev
- Filters: filter, find, not, eq
- Collections: add, end
- Tests: is

```
$('div')
  .find('a.subject')
    .click(fn)
  .end() // strip filter
  .eq(0) // like :first
    .addClass('highlight');
```

Manipulation

- Inserting: after, append, before, prepend, html, text, wrap, clone

Manipulation

- Inserting: after, append, before, prepend, html, text, wrap, clone
- Clearing: empty, remove, removeAttr

Manipulation

- Inserting: after, append, before, prepend, html, text, wrap, clone
- Clearing: empty, remove, removeAttr
- Style: attr, addClass, removeClass, toggleClass, css, hide, show, toggle

Manipulation

- Inserting: after, append, before, prepend, html, text, wrap, clone
- Clearing: empty, remove, removeAttr
- Style: attr, addClass, removeClass, toggleClass, css, hide, show, toggle

```
var a = $(document.createElement('a')); // or $('<a>')
a.css('opacity', 0.6).text('My Link').attr('href', '/home/');
$('div').empty().append(a);
```

Data

- Storing data directly against elements can cause leaks

Data

- Storing data directly against elements can cause leaks
- .data() clean way of linking data to element

Data

- Storing data directly against elements can cause leaks
- .data() clean way of linking data to element
- All handled by jQuery's garbage collection

Data

- Storing data directly against elements can cause leaks
- .data() clean way of linking data to element
- All handled by jQuery's garbage collection

```
$(this).data('type', 'forward');

var types =
$('a.subject').data('type');

$('a.subject').removeData();
```

Events

```
$('a.subject').click();
```

DOM Ready

- Most common event: DOM ready

DOM Ready

- Most common event: DOM ready

```
$(document).ready(function () {})

// or as a shortcut:

$(function () {})
```

Binding

- Manages events and garbage collection

Binding

- Manages events and garbage collection
- Event functions are bound to the elements matched selector

Binding

- Manages events and garbage collection
- Event functions are bound to the elements matched selector
- Also supports .one()

Binding

- Manages events and garbage collection
- Event functions are bound to the elements matched selector
- Also supports .one()

```
$('a.reveal').bind('click', function(event) {
  // 'this' refers to the current element
  // this.hash is #moreInfo - mapping to real element
  $(this.hash).slideDown();
}).filter(':first').trigger('click');
```

Helpers

- Mouse: click, dlbclick, mouseover, toggle, hover, etc.

Helpers

- Mouse: click, dlbclick, mouseover, toggle, hover, etc.
- Keyboard: keydown, keyup, keypress

Helpers

- Mouse: click, dlbclick, mouseover, toggle, hover, etc.
- Keyboard: keydown, keyup, keypress
- Forms: change, select, submit, focus, blur

Helpers

- Mouse: click, dlbclick, mouseover, toggle, hover, etc.
- Keyboard: keydown, keyup, keypress
- Forms: change, select, submit, focus, blur
- Windows: scroll

Helpers

- Mouse: click, dlbclick, mouseover, toggle, hover, etc.
- Keyboard: keydown, keyup, keypress
- Forms: change, select, submit, focus, blur
- Windows: scroll
- Windows, images, scripts: load, error

Custom Events

- Roll your own

Custom Events

- Roll your own
- Bind to element (or elements)

Custom Events

- Roll your own
- Bind to element (or elements)
- Trigger them later and pass data

Custom Events

- Roll your own
- Bind to element (or elements)
- Trigger them later and pass data

```
$('div.myWidget').trigger('foo', [123/*id*/]);
// id access via
// .bind('foo', function (event, id, etc) {})
```

Event Namespaces

- Support for event subsets via namespaces

Event Namespaces

- Support for event subsets via namespaces
- If you don't want to unbind all type X events - use namespaces

Event Namespaces

- Support for event subsets via namespaces
- If you don't want to unbind all type X events - use namespaces

```
$('a').bind('click.foo', fn);

$('a').unbind('.foo');
```

Ajax

```
$.ajax({ url : '/foo', success : bar });
```

Ajax Made Easy

- Cross browser, no hassle.

Ajax Made Easy

- Cross browser, no hassle.
- $.ajax does it all

Ajax Made Easy

- Cross browser, no hassle.
- $.ajax does it all
- Helpers $.get, $.getJSON, $.getScript, $.post, $.load

Ajax Made Easy

- Cross browser, no hassle.
- $.ajax does it all
- Helpers $.get, $.getJSON, $.getScript, $.post, $.load
- All Ajax requests sends: X-Requested-With = XMLHttpRequest

$.ajax

```
$('form.register').submit(function () {
  var el = this; // cache for use in success function
  $.ajax({
    url :  $(this).attr('action'),
    data : { 'username' : $('input.username').val() }, // 'this' is the link
    beforeSend : showValidatingMsg, // function
    dataType :  'json',
    type :  'post',
    success : function (data) {
      // do something with data - show validation message
    },
    error : function (xhr, status, e) {
      // handle the error - inform the user of problem
      console.log(xhr, status, e);
    }
  });
  return false; // cancel default browser action
});
```

Effects

```
$(this).slideDown();
```

Base Effects

```
$('div:hidden').show(200, fn);
$('div:visible').hide(200);
$('div').fadeIn(200);
$('div').slideDown(100);
```

Base Effects

```
$('div:hidden').show(200, fn);
$('div:visible').hide(200);
$('div').fadeIn(200);
$('div').slideDown(100);

$('div').animate({
  'opacity' : 0.5,
  'left' : '-=10px'
}, 'slow', fn)
```

Utilities

```
$.browser.version
```

Utilities

- Iterators: each, map, grep

Utilities

- Iterators: each, map, grep
- Browser versions, model and boxModel support

Utilities

- Iterators: each, map, grep
- Browser versions, model and boxModel support
- isFunction

Core Utilities

- jQuery can plays with others!

Core Utilities

- jQuery can plays with others!

```
$j = $.noConflict();
$j === $ // false
```

Core Utilities

- Extend jQuery, merge objects and create settings from defaults

Core Utilities

- Extend jQuery, merge objects and create settings from defaults

```
var defaults = { 'limit' : 10, 'dataType' : 'json' };
var options = { 'limit' : 5, 'username' : 'remy' };

var settings = $.extend({}, defaults, options);

// settings = { 'limit' : 5, 'dataType' : 'json',
'username' : 'remy' }
```

Plugins

```
$('#emails .subjects').doMagic()
```

Plugins

- Simple to write

Plugins

- Simple to write
- Makes your code *more* reusable

Plugins

- Simple to write
- Makes your code *more* reusable
- Don’t break the chain!

Simple Plugin

```
// wrap in anonymous function to use $
(function ($) {

  $.fn.log = function () {
    console.log(this);
    // returning continues the chain
    return this; // this is the jQuery collection
  };

})(jQuery);
```

More Info

Remy Sharp:

remy@leftlogic.com

leftlogic.com

remysharp.com

Resources:

jquery.com

docs.jquery.com

groups.google.com/group/jquery-en

ui.jquery.com

learningjquery.com

jqueryfordesigners.com